HONEYNET SOLUTIONS

A deployment guide

Ronald C Dodge JR, Richard T Brown, Daniel J Ragsdale

United States Military Academy

Abstract: Honeynets provide network and system managers a unique intrusion detection and monitoring system that provides indications of malicious behavior in a near "false positive" proof manner. When deployed properly, these systems can provide warning of both inside and external network threats. However if the deployment is not tightly integrated into the existing topology and the honeynet is configured to allow in only the threat intended to monitor, the usefulness will be greatly diminished. We propose a number of honeynet deployment schemes to answer some specific questions about their uses, capabilities, and resource requirements. Specifically, the paper addresses deployment of a Gen II honeynet, a solution for deploying a honeynet to a remote location, a consolidated honeynet scheme, and a model for targeted honeynets.

Key words: Honeynet, Honeyfarm, Intrusion Detection

1. INTRODUCTION

When it comes to internet-based attacks, computer network system administrators are concerned with three main principles: detection, prevention, and response. A system administrator who believes there is malicious activity on one of his networks will often consult a server, firewall, or host computer event log. Unfortunately, separating malicious connections and events from legitimate activity can be a next to impossible task – like finding a needle in a haystack. Honeynets are an extremely useful security tool because they allow system administrators to go from finding a needle in a haystack to finding a *needle in a needle stack*.

Honeynets focus on the principles of detection, response, and monitor. They are designed to be probed and attacked. All data collected is of high value because honeynets have no legitimate production traffic. The data collected is "unpolluted" by other activity. Thus, we are searching for needles in a needle stack. Honeynets are an ideally suited security tool for detecting a new internet based attack and catching advanced attackers. They will reveal the identity of the attacker, the attacker's means of communication, and the tools used in the attack. Depending on the type of attack, the signature can then be added to firewall and intrusion detection system databases or operating system software can be patched to remove a vulnerability. Ultimately, honeynets contribute to the prevention of future attacks.

Honeynets can be categorized into low and high interaction categories; we will address high interaction honeynets. In a high interaction honeynet, the server and host systems have real

operating systems and applications. These systems require extensive resources and maintenance, to include monitoring – remember, you are pacing a vulnerable system(s) on a segment of your network with the hopes it will be probed and attacked. The time required to monitor honeynets is often what makes system administrators resist using them. One of our proposed solutions includes the incorporation of a centralized honeynet system (honey farm) for large organizations to relieve the honeynet monitoring burden on the local system administrator.

The deployment of a honeynet is a risky endeavor because it has the capabilities of a full production network. Extensive risk mitigation measures must be performed to prevent a honeynet computer from being used in an attack inside your network or on outside systems. An example would be a firewall system (or as described later, a honeywall) set to drop or modify outbound packets that match a known malicious signature. Risk mitigation is essential for data control and will be discussed in this paper.

We present four deployment schemes for honeynets that solve some very specific problems. The deployment schemes are a Generation II honeynet, a solution for deploying a honeynet to a remote location, a consolidated honeynet scheme, and a model for targeted honeynets.

2. CHARACTERISTICS OF A GEN II HONEYNET

Until recently, honeynet sensors would perform bridging at layer 3 of the OSI model. Conceptually, this makes sense because the IP header contains the information necessary for routing to a honeypot system at a certain IP address. Data control was performed by a standard firewall operating at layer 3 as well. These systems operating at layer 3 are referred to as Generation I honeynets [1]. While this method of deployment works, an advanced attacker could discover in very short order that he was not on a real production system or the system was being used to monitor his actions. The potential consequences of this discovery could prove disastrous to the organization launching the honeynet. For example, a vengeful attacker could launch a distributed denial of service attack on that organization.

An important question arose from the previous scenario. Can a honeynet be deployed that is less visible to an attacker and still performs the critical tasks of data capture and data control? Gen IIhoneynets are the answer to this questionGen II honeynets have a single sensor that bridges at layer 2, as shown in figure 1. This gives two main improvements over Gen I honeynets. First, the IP header is not disturbed because Gen II honeynets operate below it. Essentially, they are routing traffic at the MAC level. This routing is invisible to an attacker because there are no TTL decrements in the IP header.

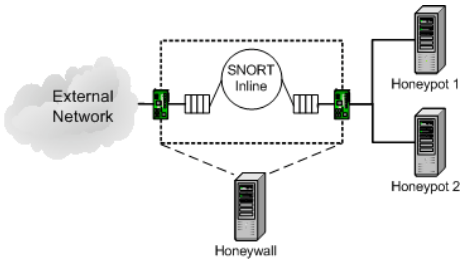

Figure 1: Honeywall Schematic

Second, a Gen II honeynets can be part of a production network. The layer 2 operations allow us to covertly insert a single honeypot into a real production network with minimal risk to the other systems in that LAN. This makes the existence of a honeypot less distinguishable to an attacker.

A honeywall performs data control for Gen II honeynets. Data control for a honeynet should allow traffic in, but carefully restrict the outbound traffic. For example, a honeywall could be configured to only allow a max of 10 outbound connections. This would protect against the machine being used for denial of service attacks. Additionally, the honeywall runs a modified version of SNORT called SNORT-Inline that utilizes the queuing feature in IPtables/EBtables to inspect every outbound packet. If the packet matches a known exploit (or any other rule the honeynet manager want to implement) one of four actions can be taken. First the packet can be allowed to pass on to the external NIC; second the packet can be dropped, third the packet can be rejected; or fourth (and more interestingly) the packet can be modified to render the exploit useless.

While the honey wall can capture all packet data in a tcpdump format and log alerts based on a snort rule set, the most valuable data is captured on the individual honeypots. If an attacker is operating in an unencrypted mode, his data can be captured using any one of several packet utilities to reassemble TCP packets. However, many attackers today encrypt their attack communication. This presents a significant challenge to data capture. One method would be to record the encrypted packets and attempt to break the encryption scheme. This can be very tedious and is not preferred. The Honeynet Project (www.honeynet.org) has developed a tool called Sebek for data capture of encrypted channels. [2]

Sebek operates on the principle that it is better to circumvent session encryption with a kernel based method rather than break the encryption. Sebek software operates at the kernel of an operating system, making it virtually invisible to an attacker. It circumvents encryption by recording an attacker's keystrokes and recording the packet stream as it is being decrypted and reassembled. Some of the information that Sebek is designed to capture or recover are files copied with SCP, passwords used to log in to remote system, passwords used to enable Burneye protected binaries, and other forensics related tasks.

The combination of using a honeywall and sebek provides a sound method of security in the honeynet system through data control and data capture. The data captured is logged and stored on a separate, secured system to prevent loss of data in case attacker is alerted to honeypot.

3. HONEYNET DEPLOYMENT SCHEMES

Honeynets can be deployed in many different ways. Each different implementation should be examined and used to meet specific security needs. In this section we present different ways that can support both economic and mission requirements.

3.1 Consolidated Honeynets (Honey Farms)

System administrators often have too little time and resources to devote to managing a honeynet. An organization with extremely limited time and resources for information security should focus on intrusion detection systems, firewalls, and sound security practices first. In general, honeypots should only be used in conjunction with other security mechanisms. Although the local system administrator may not have the time to dedicate to a honeynet, the benefits of honeynet deployment are far too great to let fall to the chopping block. A distributed organization can consolidate its honeynet traffic in one physical location that is monitored by dedicated professionals 24/7. This concept is called a honey farm, shown in Figure 2.

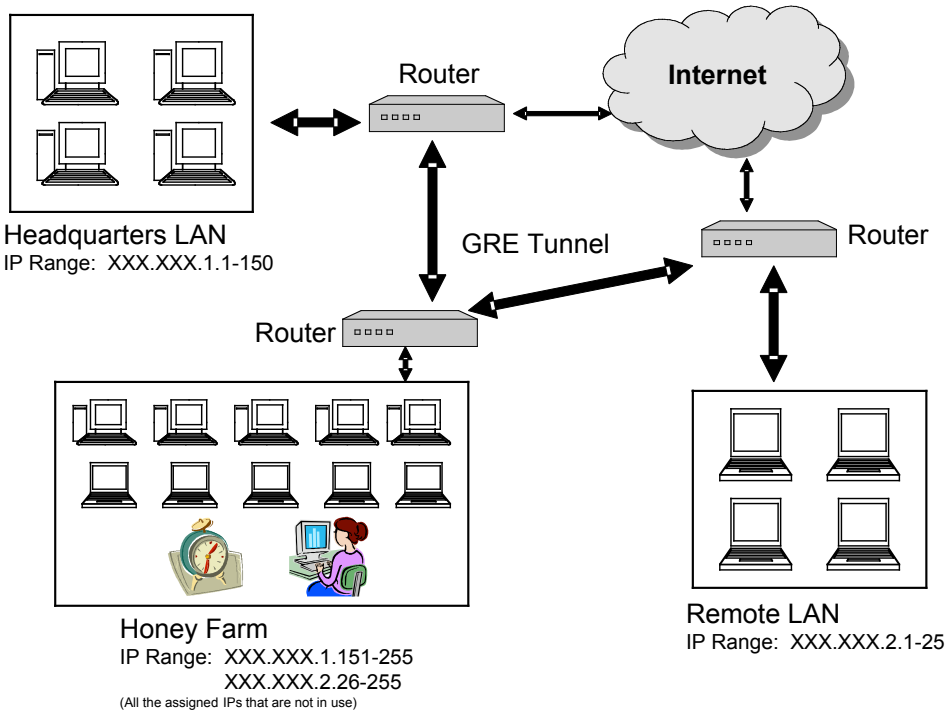

Figure 2: Model of a consolidated honeynet with a honey farm.

Honey farms consolidate all honeynet traffic by becoming the destination for scanned IP addresses that are owned by an organization but not in use in a real production system. All honeynet traffic is redirected from a router outside of a production system to a router outside of the honey farm though a Generic Routing Encapsulation (GRE) tunnel. Cisco developed this protocol to allow an arbitrary network protocol A to be transmitted over any other arbitrary network protocol B, by encapsulating the packets of A within GRE packets, which in turn are contained within packets of B [3]. A GRE tunnel achieves many of the same advantages of the routing done by a Gen II honeynet. Both protocols operate at layer 2. A GRE tunnel, through its encapsulation, encloses the IP header of the original packet and tricks the TCP/IP protocol into only performing one TTL decrement of the payload packet. When applied in the honey farm scenario, there is no decrement of the TTL of payload packet until it arrives in the honey farm where the actual IP address being attacked is physically located.

In addition to data control and data capture, honey farms perform the additional role of data collection. All the honeypot computers for a distributed organization are physically located together making collection and analysis of the data an easy endeavor.

A very robust application of the honey farm for an enterprise involves dynamically reconfiguring each router to account for differing threat levels across an enterprise. An enterprise located in three different countries may experience a sudden increase in the number of scans and attempted attacks from one of the countries. By dynamically configuring the routers, an enterprise can have its routers assign more of the honeynet IP addresses to the country under attack. This will enhance the attack data received and give a greater probability of capturing a new attack.

3.2 Deployable Solution to a Remote Area

In today's global mobility, an organization will often set up a new base of operations in a foreign country. It is necessary to network operations in that new "remote" location to the headquarters and this is usually accomplished over the Internet. While global expansion is usually good for business, there are some risks in the modern world of terrorism and hackers. Any organization (commercial business, government, military, humanitarian, etc.) must ask itself an important question – "Am I being specifically targeted based on my deployment or expansion into a new geographic area?" A deployable honeynet can be a great tool to help answer this question.

Initially, an organization will set up a LAN with a small footprint at its remote location. The remote site can incorporate a small honeynet or virtual honeynet using a laptop with technology such as UML, HoneyD, or virtual machines. If the honeynets at the remote site and back at the headquarters capture the same types of attacks from the same SRC, there is good reason to believe that the organization is being specifically targeted. Specific targeted computer attacks are usually performed by an advanced attacker. It is for this reason that a high interaction honeynet vice a low interaction honeypot should be incorporated at both sites.

The organization may want to incorporate the consolidated honeynet technology from the previous section. The system administrator at the remote site will probably not have the time to constantly monitor and conduct analysis of the data captured from the honeynet. A GRE tunnel can be implemented from the remote honeynet back to a honey farm at the headquarters (see Figure 2).

3.3 Targeted Honeynets

The deployment of honeynets involves more than a network topology template. The identification of the type of malicious activity that is desired to be observed is integral to the effectiveness of a honeynet. The concept of a honeynet is that all data observed on the honeynet is suspicious. So the normal activity of examining IDS logs looking for malicious activity within normal traffic is likened to looking for a needle in a haystack; a honeynet turns this into looking for a needle in a needle stack. The idea of a "tuned honeynet" is to reduce the number of needles. As an example; the security level, topology, and content is significantly different if your target is a "script kiddie" than if it is a sophisticated intruder. This can be predicated on specific intelligence on the potential intruder, although it is not required. If a certain subnet of an organizations network has been compromised and an expectation exists that the attacker might continue to other network segments, then a honeynet can be established that presents a very similar configuration as the one the attacker already compromised. If no confirmed intrusions have been documented, a honeynet can be established to present to an attacker the activity that is most critical to your organization. For example, for an organization that has valuable proprietary data, the honeynet would be constructed to

- Be fairly secured. Lesser skilled attackers would not be interested in the organizations data – they are more interested in credit card accounts, hard drive space, and installing relay bots. The security level should be high enough to "screen out" these attackers. Attackers with a focus on state or corporate sponsored espionage should be able to bypass most normal security techniques.
- Be a believable topology. The topology should not present just a single machine on a subnet. Populate the honeynet with services and systems that mimic a real deployment. As an example, the topology might place a file server on the same subnet of a vulnerable web server. This subnet could be placed behind a firewall with an open port that an attacker could use.

- • The content of the web and file server should be believable. A honeynet without honey will provide only limited information about the attacker.

Sometimes we want to create a honeynet that is designed to catch a certain kind of attacker. The level of security required (S, S', S'', etc) depends on the threat and the data being "protected." Some examples are:

- S -- Script Kiddie – P2P Server (Kazaa, etc)
- S' -- Credit Card DB
- S' -- Military Troop Info
- S'' -- Chem-Bio Warfare Info

4. CONCLUSION

Honeynet use is rapidly increasing. In May 2004 the Honeynet Project released a honeywall CD Rom that was downloaded over 150,000 times in a two week period. Honeynets are capable of capturing attack data that may otherwise go unnoticed. The deployment of honeynets should be focused on the threat that an organization is interested in tracking. As honeynet use continues to grow, attackers will surely develop ways to detect their use. Operating at layer 2 provides a method of hiding the presence of the honeynet sensor, but this will only be temporary until attackers find a way to detect layer 2 routing. Security professionals everywhere should constantly strive to invent new innovative ways to outperform the attacker community. Ways to mitigate the resources required to implement honeynet solutions include honeyfarms and deployable "honeynet in a box" solutions that need only be connected to a network.

5. REFERENCES

[1] Know Your Enemy: Honeynets, 12 November 2003. Retrieved off the world wide web from the Honeynet Project www.honeynet.org/.

[2] Know Your Enemy: Sebek, 17 November 2003. Retrieved off the world wide web from the Honeynet Project www.honeynet.org/.

[3] Spitzner, Lance. Honeypots: Tracking Hackers, Pearson Education, Inc., 2003.